connect • monitor • manage

CONNECT - APPLICATION NOTE

Configuring Windows Server 2012 R2 with NVGRE using the Emulex OCe14102 Adapter

Emulex provides Network Interface Card (NIC) Ethernet driver support for Windows Server 2012 R2. The Emulex OCe14102 adapter supports Network Virtualization using Generic Encapsulation (NVGRE) offloads to support a multi-tenant cloud infrastructure and to eliminate the 4096 virtual LAN (VLAN) limit.

Solution Implementer's Series™

Emulex OneConnect® Network Adapters

EMULEX®

Introduction

This application note presents a step-by-step guide of a simple NVGRE configuration on Windows Server 2012 R2 using the Emulex OCe14102 adapter. Configuration and deployment of NVGRE is often perceived as difficult, however, this application note provides the necessary hardware, software requirements and the steps for a successful deployment using the Emulex Oce14102 adapter.

Note: This application note does not give any details on performance matrix for NVGRE or configuration details when implementing NVGRE with gateways.

The proof of concept (POC) can divided into two phases.

Phase 1 – Two hosts connected to 10Gbps networking switch. This is scalable solution when more hosts are added.

Phase 2 – Two hosts connected back-to-back. This is a solution when a switch is not available for the infrastructre.

Below are the two configurations used for the purpose of validating NVGRE in Emulex Labs.

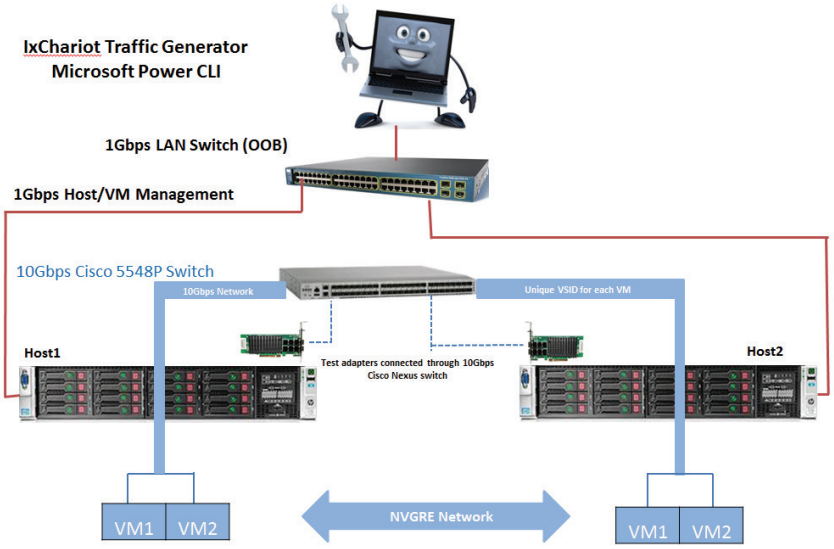

Figure 1. Two hosts connect to 10Gbps networking switch.

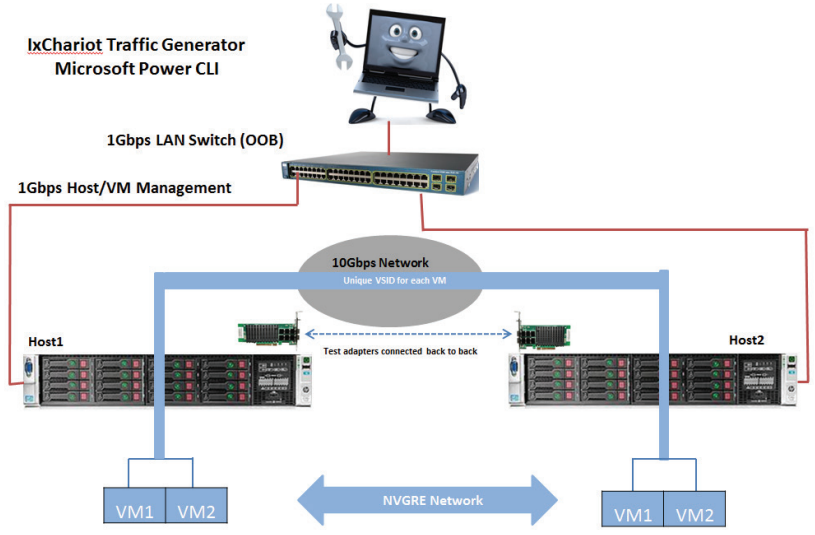

Figure 2. Host to Host connected back-to-back.

Hardware requirements

Hardware components	Quantity	Description	POC components
Server	2	Any server with Intel/AMD processors which supports Windows 2012 R2	HP DL 380p Gen8 with Intel® Xeon® CPU E5-2690 @ 2.90GHz
Hard drives	2	Any SAS or SATA drive	SAS 300GB drives
RAID controller	2	Any server with supports Windows 2012 R2	
RAM	12-96GB/server		96GB RAM/server
OCe14102	2	Emulex 10GbE adapter with NVGRE offload	OCe14102
Switch	1	10Gbps switch	Cisco 5548P
Cables	2	10Gbps optical SFP+ cables	10Gbps optical SFP+ cables

Software requirements

Component	Quantity	Description	POC components
Windows 2012 R2 Server or DataCenter OS	2	License key and Windows 2012 R2 OS	Windows 2012 R2 DataCenter
Wireshark	1	Free Packet capturing and analyzing tool for verification	Wireshark installed on one of the hosts
I/O Tools a) Iometer b) NTttcp c) IxChariot d) Medusa Labs Test Tools e) Any other IO tool available within your organization	2	Iometer, NTttcp are free IO tools which can be used for verification. IxChariot and Medusa Labs Test Tool requires a license.	Licensed IxChariot tool

Pre-installation

There are several steps to consider before applying power or installing an operating system on any system. First, you need to ensure rack space and appropriate power and cooling are available. A key step is to ensure all hardware and software is at the latest firmware levels and download any necessary patches and drivers.

As a best practice, updating the servers and converged network adapters to the latest firmware and drivers. For other components such as LAN on motherboard (LOM) and integrated management ports, please refer to the server vendor website or user's manual for the latest firmware and driver.

The Emulex OCe14102 network adapter used the updated firmware and Windows 2012 R2 NIC driver. Please refer to Emulex for the latest firmware and driver.

Post-installation

After installing Windows 2012 R2 Server or DataCenter on both the hosts along with Hyper-V role, name the hosts. For this POC, the hosts have been named as "nvgre1" and "nvgre2". Install at least two virtual machines (VMs) on each host. For this POC, the VMs have been named as "Red1" and "Blue1" on nvgre1 and "Red2" and "Blue2" on nvgre2.

NVGRE offload should be enabled on the Emulex OCe14102 adapters. For verifying if the offload is enabled go to the OCe14102 adapter's Advanced Properties and verify the value to be "Enabled."

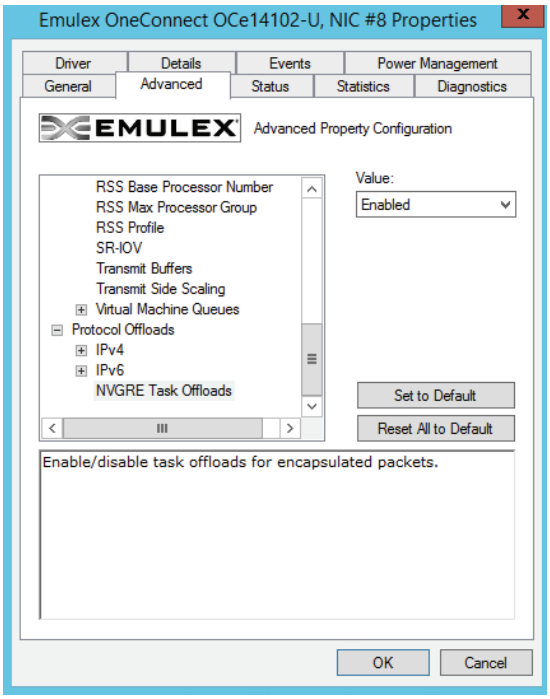

Figure 3. Advanced Properties for Emulex OCe14102.

Phase 1: Two hosts connect to 10Gbps networking switch

Implementation/Configuration using an Emulex OCe14102 adapter

This POC was created using the modified version of simple and scalable multi-tenant topology demo script by Microsoft for Windows 2012. The script was modified for Windows 2012 R2. The key objective was to describe the configuration and validation of this multi-tenant topology with NVGRE.

Step 1 – Configure the OCe14102's IP address on Host1 and Host2

This section enlists the detailed steps followed in this POC for the configuring IP address on the OCe14102 adapter after installing the latest firmware and driver.

a) On "nvgre1," go to the Server Manager; select Local Server from the console tree. Click the link next to EthernetX in the Properties tile.

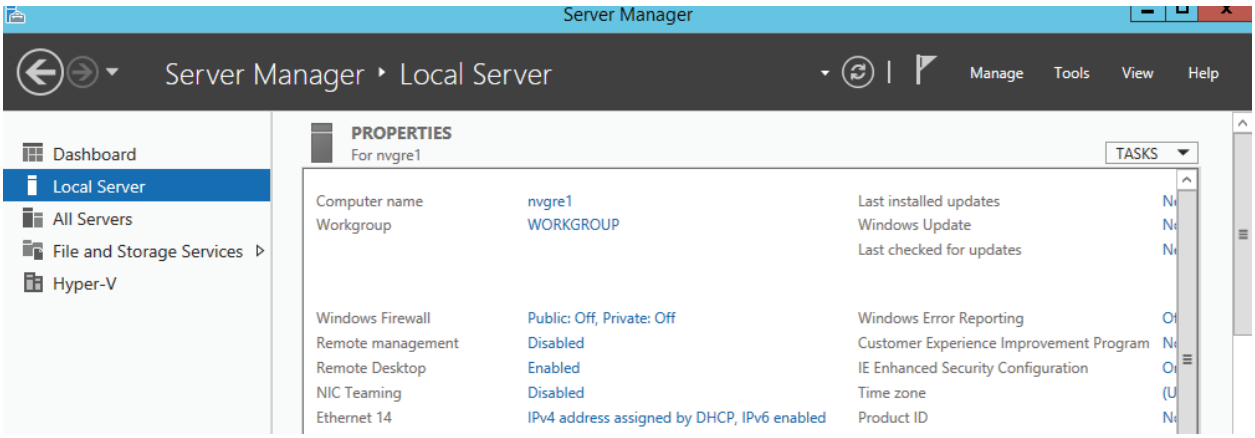

Figure 4. Server Manager.

b) In the Network Connections window, right-click EthernetX, and then click Properties.

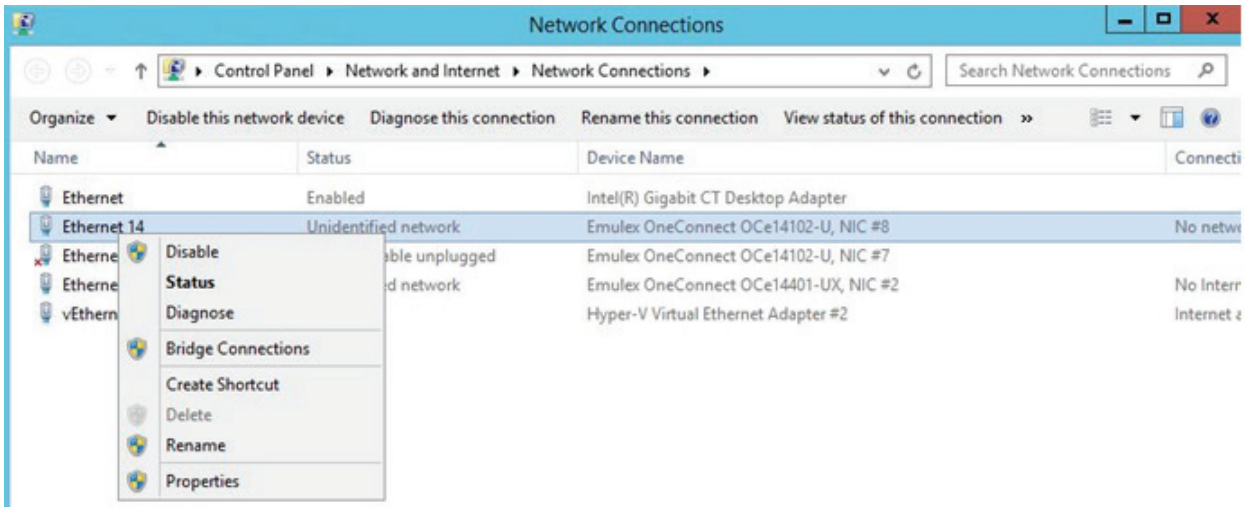

Figure 5. Network connections.

c) Click Internet Protocol Version 4 (TCP/IPv4), and then click Properties.

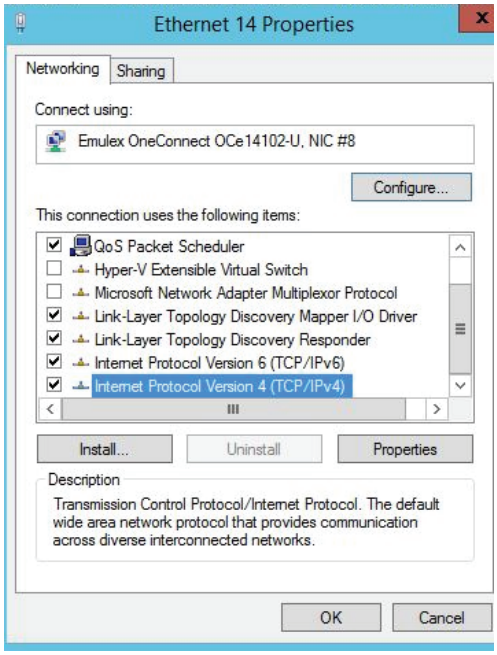

Figure 6. Ethernet properties.

d) Assign the IP address as 192.168.4.11 and Subnet mask as 255.255.255.0 and click OK.

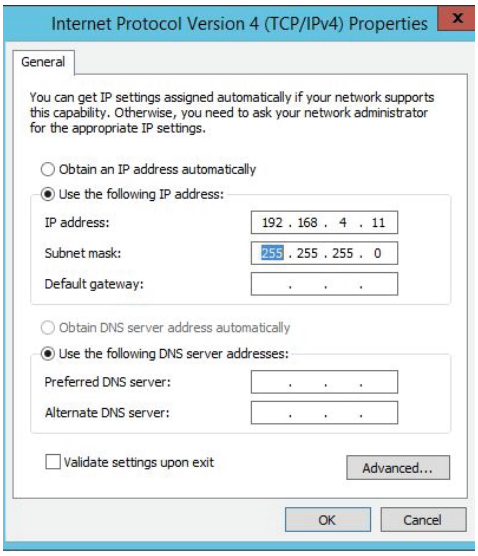
Internet Protocol Version 4 (TCP/IPv4) Properties
General
You can get IP settings assigned automatically if your network supports this capability. Otherwise, you need to ask your network administrator for the appropriate IP settings.
Obtain an IP address automatically
Use the following IP address:
IP address: 192 . 168 . 4 . 11
Subnet mask: 255 . 255 . 255 . 0
Default gateway:
Obtain DNS server address automatically
Use the following DNS server addresses:
Preferred DNS server:
Alternate DNS server:
Validate settings upon exit
Advanced...
OK
Cancel

Figure 7. TCP/IPv4 Properties.

e) Repeat the steps a through d on "nvgre2": assign the IP address as 192.168.4.22 and the Subnet mask as 255.255.255.0.

Step 2 – Modify or disable the firewall

This section enlists the step to disable the firewall. However, this might not be the case in a production enviroment. Depending on the policy, please enable ping, I/O tool services and any other services required for enabling NVGRE policies.

a) Modfy or disable the host firewall according to the organization's security policy.

b) For this POC, the firewall has been disabled on both the hosts.

```
PS C:\Users\Administrator\Desktop\NvgrePolicies> netsh advfirewall set AllProfiles state off
Ok.
```

Figure 8. Example to disable firewall.

Step 3 – Configure a virtual switch on Host1 and Host2

This section enlists the steps to create, configure and assign a new virtual switch to the VMs

a) On "nvgre1," from Server Manager, click the "Tools" and click "Hyper-V Manager."

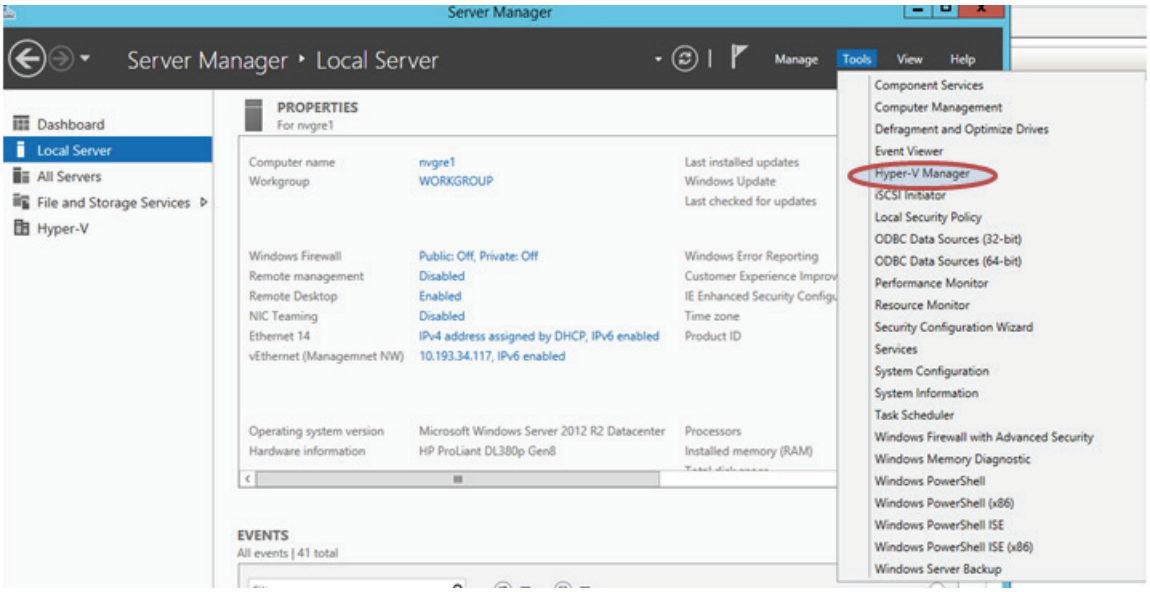

Figure 9. Hyper-V Manager.

b) Select the Virtual Switch Manager from the Actions pane.

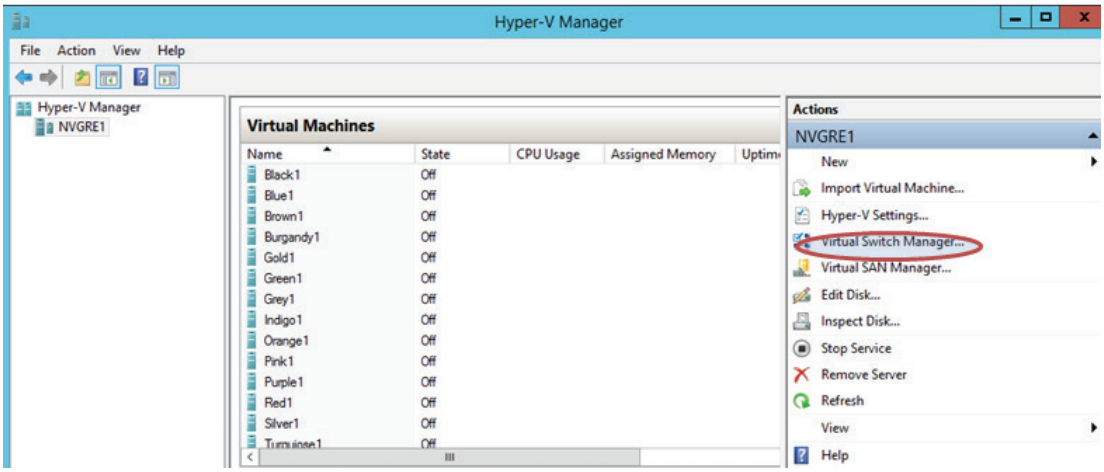

Figure 10. Virtual switch Manager.

c) Create a new virtual switch by selecting "new virtual network switch." Make sure "External" is selected for type of switch and click OK.

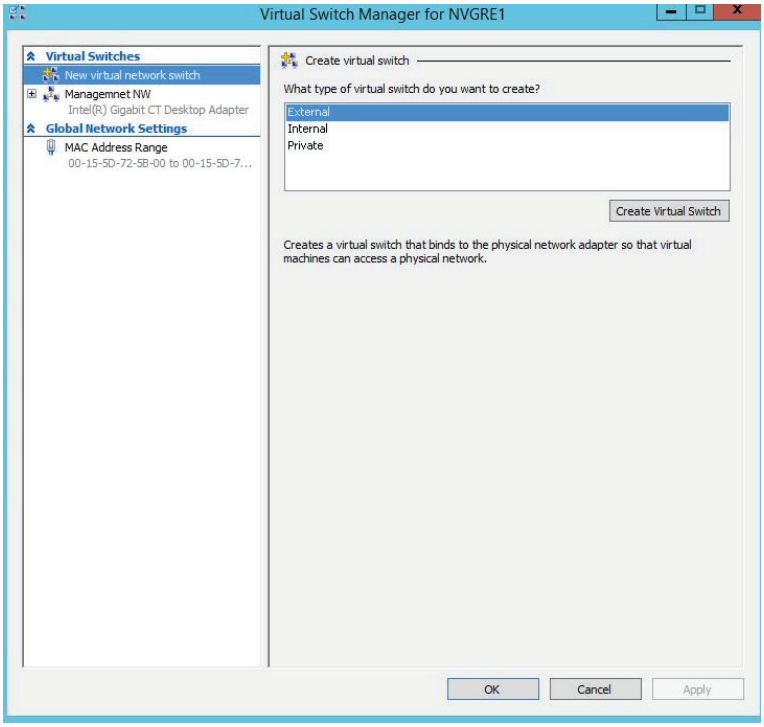

Figure 11. Create a new virtual switch.

d) Name the virtual switch "NVGRE" and select the Emulex OCe14102 adapter from the drop-down menu of the external network. Make sure the options "Allow management operating system to share this network adapter" and "Enable single-root I/O virtualization" are not selected. Click on Apply.

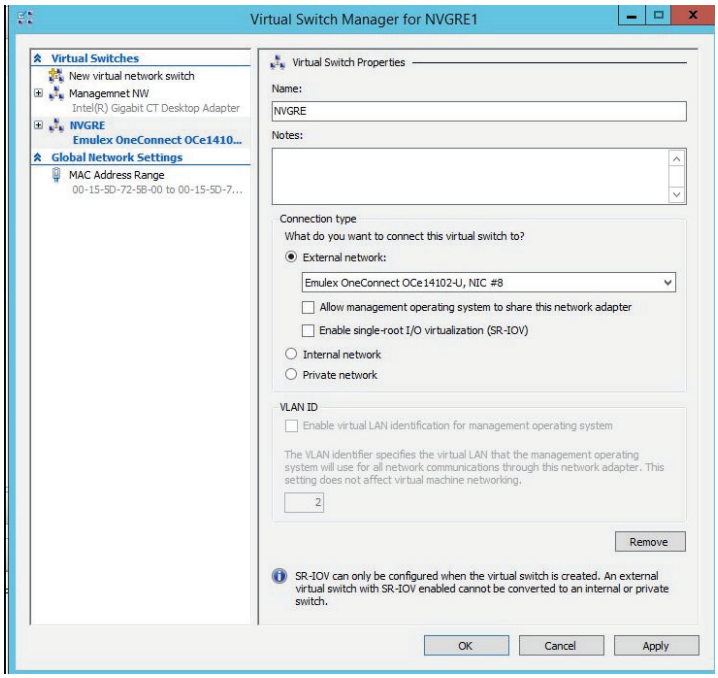
Virtual Switch Manager for NVGRE1
Virtual Switches
New virtual network switch
Managemnet NW
Intel(R) Gigabit CT Desktop Adapter
NVGRE
Emulex OneConnect OCe1410...
Global Network Settings
MAC Address Range
00-15-5D-72-5B-00 to 00-15-5D-7...
Virtual Switch Properties
Name:
NVGRE
Notes:
Connection type
What do you want to connect this virtual switch to?
External network:
Emulex OneConnect OCe14102-U, NIC #8
Allow management operating system to share this network adapter
Enable single-root I/O virtualization (SR-IOV)
Internal network
Private network
VLAN ID
Enable virtual LAN identification for management operating system
The VLAN identifier specifies the virtual LAN that the management operating system will use for all network communications through this network adapter. This setting does not affect virtual machine networking.
2
Remove
SR-IOV can only be configured when the virtual switch is created. An external virtual switch with SR-IOV enabled cannot be converted to an internal or private switch.
OK
Cancel
Apply

Figure 12. Properties for the virtual switch.

e) Repeat steps a through d on "nvgre2."

Step 4 – Assign the network adapter for each VM and assign appropriate IP address to the interface for each VM.

a) Make sure the VM on "nvgre1" and "nvgre2" are turned OFF.

b) Click "Settings" from the Actions pane for each VM for "nvgre1."

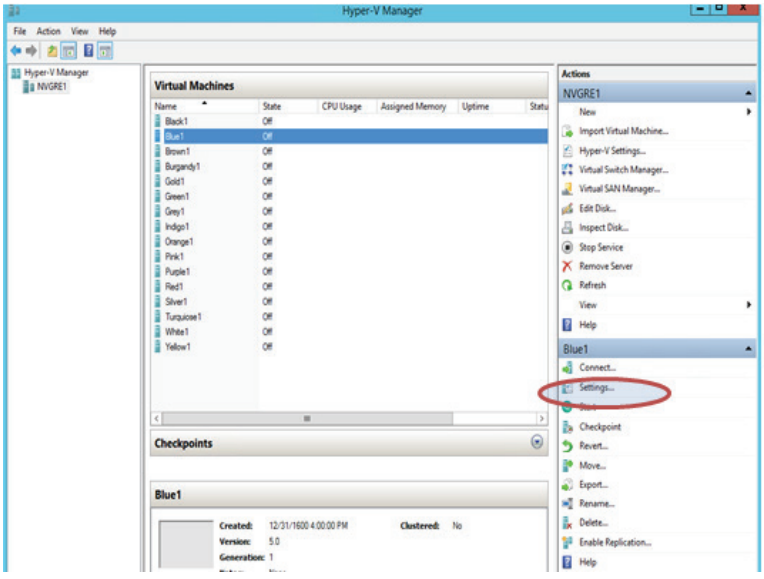

Figure 13. VM settings.

c) Under "Add Hardware" select "Network Adapter" and click on Add.

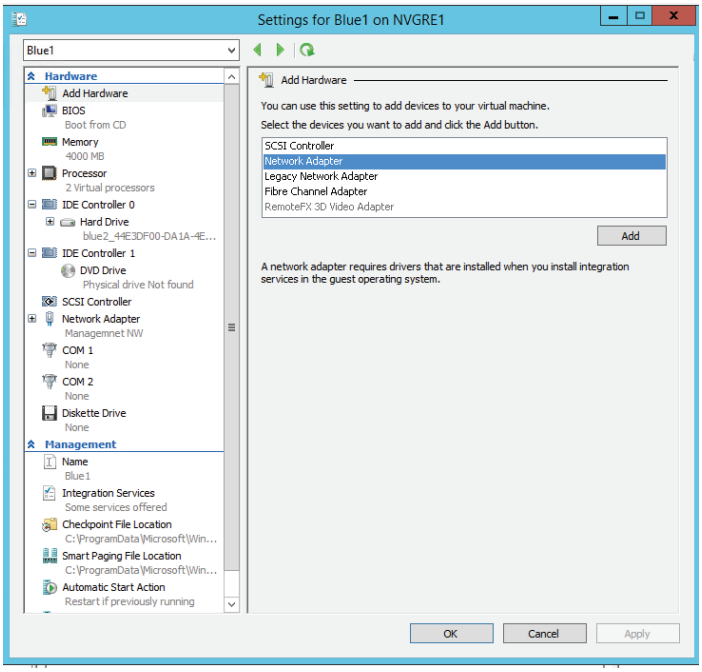

Figure 14. Add hardware.

d) Select the NVGRE Virtual Switch from the drop-down menu.

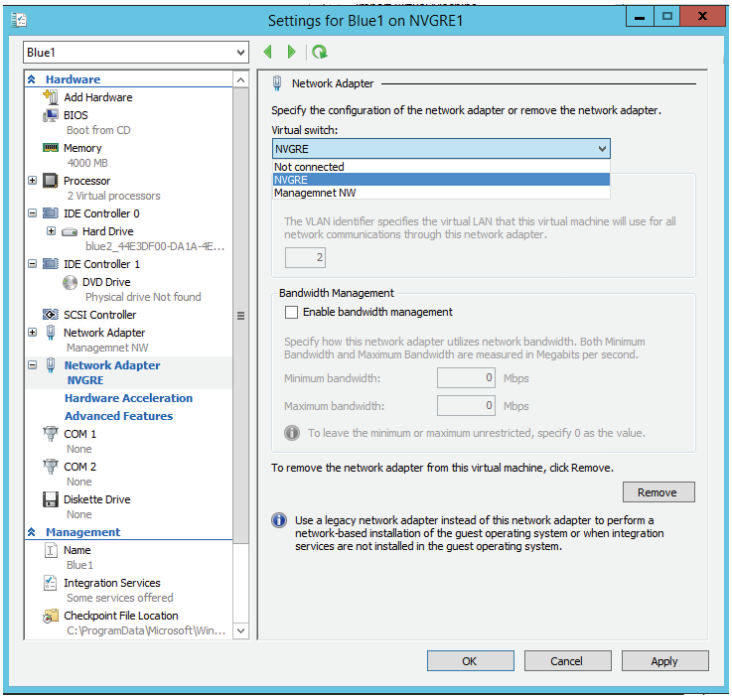

Figure 15. Add NVGRE as the new network adapter for each of the VM.

e) Expand the Network Adapter option and check for "Enable VMQ" under Hardware Acceleration.

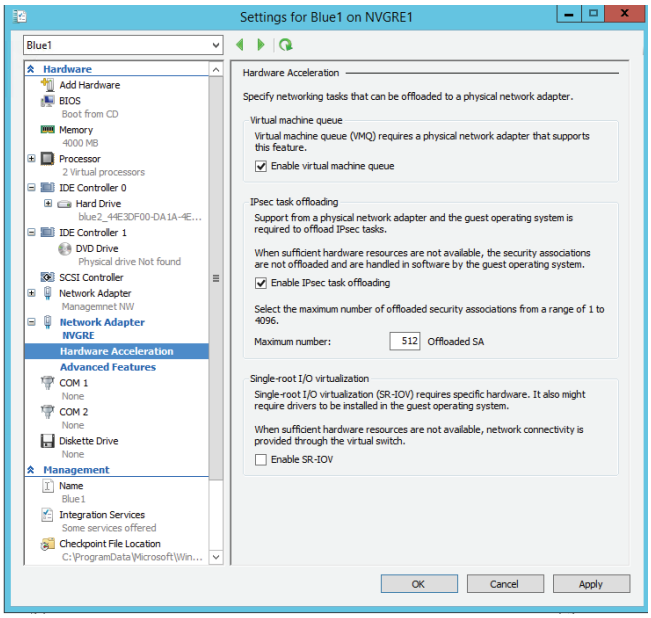

Figure 16. Enable VMQ.

f) Assign static MAC address under "Advanced Features." Note that the Dynamic MAC address can be assigned too. Make sure each VM has a unique MAC address on "nvgre1" and "nvgre2."

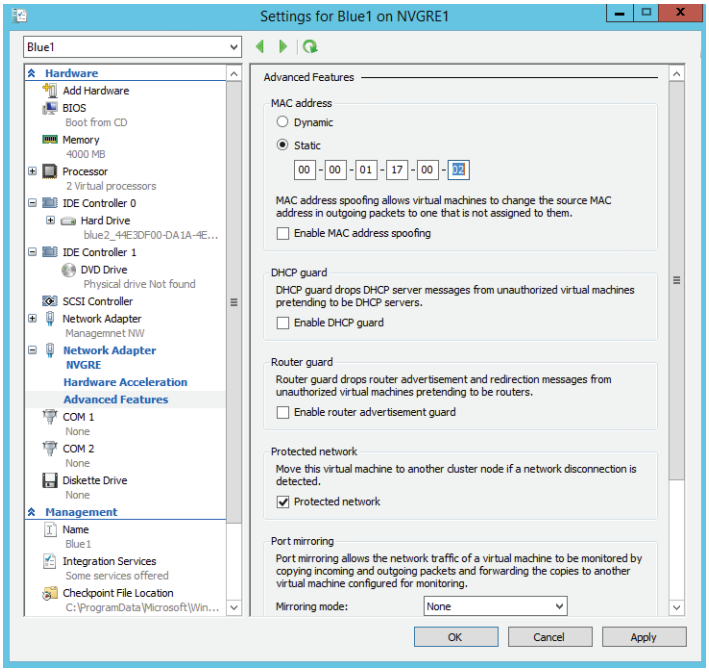

Figure 17. Assign MAC address.

g) Turn ON each VM and assign an IP address to the interface using Network Manager or using netsh command.

Blue1: 10.0.0.3
Red1: 10.0.0.13

Note that the VM IP addresses can be the same as they are uniquely identified by VSID.

```
netsh interface ip set address "Ethernet 7" static 10.0.0.3 255.255.255.0
```

Figure 18. Set up IP addresses on the VM.

h) Repeat steps a through g for "nvgre2". Use the following IP's for VM on "nvgre2:"

Blue1: 10.0.0.9
Red1: 10.0.0.9

NOTE: For this POC, a separate 1Gpbs Management network and virtual switch was setup for remote access and for internet connectivity purpose using the steps mentioned above.

Step 5 – Test the network on the VM.

This section enlists the basic steps for validating the VM network before applying any NVGRE policies.

a) Ping Blue2 from Blue1 and Red1 from Red2 to ensure that the VM network is pingable.

b) If you cannot ping, make sure there is no firewall policies set on the host or the VM.

Step 6 – Create powershell scripts to create lookup records.

This section enlists the details for the command to create lookup record policy entry for an IP address which belongs to a Microsoft® Hyper-V® Server 2012 R2 virtual network.

a) Microsoft has a sample powershell script to create the NVGRE policy for Windows 2012.

However for this POC, as mentioned before, the script was modified for Windows 2012 R2.

b) Lookup records need to be created for each VM on every Hyper-V host as it creates a policy entry for an IP address in a virtual network.

c) The command to create lookup records is "New-NetVirtualizationLookupRecord".

Below are the listed parameters:

- CustomerAddress: Specifies IP address of the VM.
- ProviderAddress: Specifies IP address for a physical address that corresponds to the Customer Address.
- VirtualSubnetID: Specifies numerical value representing the unique network.
- MACAddress: Specifies MAC address of the VM guest without any ":". For example: "000001170002."

Use "Get-VMNetworkAdapter -VMName *" command to get all the MAC addresses for the VM

```
PS C:\Users\Administrator\Desktop\NvgrePolicies> Get-VMNetworkAdapter -VMName *

Name            IsManagementOs VMName     SwitchName    MacAddress   Status IPAddresses
----            -------------- ------     ----------    ----------   ------ -----------
Network Adapter False          Yellow1    Managemnet NW 00155D725B19        {}
Network Adapter False          Yellow1    NVGRE         000001170016        {}
Network Adapter False          White1     Managemnet NW 00155D725B1B        {}
Network Adapter False          White1     NVGRE         000001170015        {}
Network Adapter False          Turquiose1 Managemnet NW 00155D725B1F        {}
Network Adapter False          Turquiose1 NVGRE         000001170014        {}
Network Adapter False          Silver1    Managemnet NW 00155D725B1D        {}
Network Adapter False          Silver1    NVGRE         000001170013        {}
Network Adapter False          Red1       Managemnet NW 00155D725B11 {Ok}   {10.193.34.118, fe80::28d1:afda:638e:fe5b}
Network Adapter False          Red1       NVGRE         000001170012 {Ok}   {10.0.0.13, fe80::5cc0:e2b7:ff2d:3d9}
Network Adapter False          Purple1    Managemnet NW 00155D725B15        {}
Network Adapter False          Purple1    NVGRE         000001170011        {}
Network Adapter False          Pink1      Managemnet NW 00155D725B16        {}
Network Adapter False          Pink1      NVGRE         000001170010        {}
Network Adapter False          Orange1    Managemnet NW 00155D725B17        {}
Network Adapter False          Orange1    NVGRE         000001170009        {}
Network Adapter False          Indigo1    Managemnet NW 00155D725B1C        {}
Network Adapter False          Indigo1    NVGRE         000001170008        {}
Network Adapter False          Grey1      Managemnet NW 00155D725B18        {}
Network Adapter False          Grey1      NVGRE         000001170007        {}
Network Adapter False          Green1     Managemnet NW 00155D725B12        {}
Network Adapter False          Green1     NVGRE         000001170006        {}
Network Adapter False          Gold1      Managemnet NW 00155D725B1E        {}
Network Adapter False          Gold1      NVGRE         000001170005        {}
Network Adapter False          Burgandy1  Managemnet NW 00155D725B2C        {}
Network Adapter False          Burgandy1  NVGRE         000001170004        {}
Network Adapter False          Brown1     Managemnet NW 00155D725B14        {}
Network Adapter False          Brown1     NVGRE         000001170003        {}
Network Adapter False          Blue1      Managemnet NW 00155D725B0F {Ok}   {10.193.34.119, fe80::6ca2:c33d:d4d1:7092}
Network Adapter False          Blue1      NVGRE         000001170002 {Ok}   {169.254.80.184, fe80::544:376:9cad:50b8}
Network Adapter False          Black1     Managemnet NW 00155D725B1A        {}
Network Adapter False          Black1     NVGRE         000001170001        {}
```

Figure 19. Sample output for "Get-VMNetworkAdapter-VMName *."

- Rule: Specifies which type of virtualization mechanism is being used. Use "TranslationMethodEncap" for NVGRE.

d) Create a powershell script named "Addpolicies_2vm.ps1" on "nvgre1" and "nvgre 2" and add the lines listed below to add the Lookup record for VMs on "nvgre1" and "nvgre2:"

```
New-NetVirtualizationLookupRecord -VirtualSubnetID "5001" -CustomerAddress "10.0.0.3"
-ProviderAddress "192.168.4.11" -MACAddress "000001170002" -Rule "TranslationMethodEncap"

New-NetVirtualizationLookupRecord -VirtualSubnetID "5001" -CustomerAddress "10.0.0.9"
-ProviderAddress "192.168.4.22" -MACAddress "000002170002" -Rule "TranslationMethodEncap"

New-NetVirtualizationLookupRecord -VirtualSubnetID "6001" -CustomerAddress "10.0.0.13"
-ProviderAddress "192.168.4.11" -MACAddress "000001170012" -Rule "TranslationMethodEncap"

New-NetVirtualizationLookupRecord -VirtualSubnetID "6001" -CustomerAddress "10.0.0.19"
-ProviderAddress "192.168.4.22" -MACAddress "000002170012" -Rule "TranslationMethodEncap"
```

Step 7 – Add customer routes

This section enlists the details for the command to create a virtual network route in Microsoft Hyper-V Server 2012 R2 virtual network.

a) The customer route must to be added on each Hyper-V host as it creates a virtual network route.

b) The command for adding customer route is "New-NetVirtualizationCustomerRoute." Below are the parameters:

- DestinationPrefix: Specifies destination VM address prefix.
- NextHop: Specifies IP address for next hop gateway for the route.
- RoutingDomainID: Specifies a unique identifier for the routing domain. It allows multiple virtual subnets.
- Metric: Specifies an integer value for the route.
- VirtualSubnetID: Specifies numerical value representing the unique network.

Note: Use the same value used in the previous add lookup record section.

c) Add the lines listed below to "Addpolicies_2vm.ps1"script on "nvgre1" and "nvgre 2" to add the customer route:

```
New-NetVirtualizationCustomerRoute -RoutingDomainID "{11111111-2222-3333-4444-000000005001}"
-VirtualSubnetID "5001" -DestinationPrefix "10.0.0.0/24" -NextHop "0.0.0.0" -Metric 255

New-NetVirtualizationCustomerRoute -RoutingDomainID "{11111111-2222-3333-4444-000000006001}"
-VirtualSubnetID "6001" -DestinationPrefix "10.0.0.0/24" -NextHop "0.0.0.0" -Metric 255
```

Step 8 – Add Provider Address and Provider Route

This section enlists the details for the command to assign a Provider Address to a network interface for use in Microsoft Hyper-V Server 2012 R2 virtual network.

a) The provider address and provider route must be added, as it is unique to each host. It assigns a provider address to a network interface and creates a provider route in the virtual network

b) The command for adding the provider address is "New-NetVirtualizationProviderAddress." Below are the parameters:

- InterfaceIndex: Specifies the interface index of the physical interface (OCe14102).

  Get-netadapter commands gives the Interface Index.

```
PS C:\Users\Administrator\Desktop\NvgrePolicies> Get-NetAdapter

Name                      InterfaceDescription                ifIndex Status       MacAddress             LinkSpeed
----                      --------------------                ------- ------       ----------             ---------
vEthernet (Managemnet NW) Hyper-V Virtual Ethernet Adapter #2      29 Up           68-05-CA-1A-FC-40       10 Gbps
Ethernet 14               Emulex OneConnect OCe14102-U, NIC #8     34 Up           00-90-FA-6A-FB-F6       10 Gbps
```

Figure 20. Sample output for "Get-netadapter".

- ProviderAddress: Specifies the IP address configured for the OCe14102 adapter.
- PrefixLength: Specifies the length of the IP prefix.

c) Add the lines listed below to "Addpolicies_2vm.ps1" script on "nvgre1" to add the Provider Address. Make sure to update the Interface Index according to the output on the system:

```
New-NetVirtualizationProviderAddress -InterfaceIndex 34 -ProviderAddress 192.168.4.11
-PrefixLength 24
```

d) Add the lines listed below to "Addpolicies_2vm.ps1" script on "nvgre2" to add the Provider Address. Make sure you update the Interface Index according to the output on the system:

```
New-NetVirtualizationProviderAddress -InterfaceIndex 24 -ProviderAddress 192.168.4.22
-PrefixLength 24
```

e) The command for adding Provider Route is "New-NetVirtualizationProviderRoute". Below are the parameters:

- InterfaceIndex: Specifies the interface index of the physical interface (OCe14102).

  Get-netadapter command gives the Interface ID.

```
PS C:\Users\Administrator\Desktop\NvgrePolicies> Get-NetAdapter

Name                      InterfaceDescription                ifIndex Status       MacAddress             LinkSpeed
----                      --------------------                ------- ------       ----------             ---------
vEthernet (Managemnet NW) Hyper-V Virtual Ethernet Adapter #2      29 Up           68-05-CA-1A-FC-40       10 Gbps
Ethernet 14               Emulex OneConnect OCe14102-U, NIC #8     34 Up           00-90-FA-6A-FB-F6       10 Gbps
```

Figure 21. Sample output for "Get-netadapter".

- DestinationPrefix: Specifies destination VM address prefix
- NextHop: Specifies IP address for next hop gateway for the route
- Metric: Specifies an integer value for the route.

f) Add the lines listed below to "Addpolicies_2vm.ps1" script on "nvgre1" to add the Provider route. Make sure to update the Interface Index according to the output on the system:

```
New-NetVirtualizationProviderRoute -DestinationPrefix "10.0.0.0/24" -InterfaceIndex 34 -NextHop
"0.0.0.0" -Metric 255
```

g) Add the lines listed below to "Addpolicies_2vm.ps1" script on "nvgre2" to add the provider route. Make sure to update the Interface Index according to the output on the system:

```
New-NetVirtualizationProviderRoute -DestinationPrefix "10.0.0.0/24" -InterfaceIndex 24 -NextHop
"0.0.0.0" -Metric 255
```

Step 9 – Add VirtualSubnetID on the VMs' network port

This section enlists the details for assigning an unique VirtualSubnet ID for each VM.

a) Add the lines listed below to "Addpolicies_2vm.ps1" script on "nvgre1" to add the virtualSubnetID to VM network port:

```
Get-VMNetworkAdapter "Blue1" | where {$_.MacAddress -eq "000001170002"} | Set-VMNetworkAdapter
-VirtualSubnetID 5001

Get-VMNetworkAdapter "Red1" | where {$_.MacAddress -eq "000001170012"} | Set-VMNetworkAdapter
-VirtualSubnetID 6001
```

b) Add the lines listed below to "Addpolicies_2vm.ps1" script on "nvgre2" to add the virtualSubnetID to VM network port:

```
Get-VMNetworkAdapter "Blue2" | where {$_.MacAddress -eq "000002170002"} | Set-VMNetworkAdapter
-VirtualSubnetID 5001;

Get-VMNetworkAdapter "Red2" | where {$_.MacAddress -eq "000002170012"} | Set-VMNetworkAdapter
-VirtualSubnetID 6001;
```

Step 10 – Run the script and verify the configuration

This section enlists the details running the script on each host and validating if the policy is set correctly.

a) Run the script on "nvgre1" and "nvgre2."

b) Run the following commands on "nvgre1" and "nvgre2" and make sure all the MAC addresses, IP addresses, VSID, Customer Route, Provider Route are set correctly.

```
Get-NetVirtualizationProviderAddress
```

```
PS C:\Users\Administrator\Desktop\NvgrePolicies> Get-NetVirtualizationProviderAddress


ProviderAddress   : 192.168.4.11
InterfaceIndex    : 34
PrefixLength      : 24
VlanID            : 0
AddressState      : Preferred
MACAddress        : 0090fa6afbf6
ManagedByCluster  : False
```

Figure 22. Sample output for "Get-NetVirtualizationProviderAddress."

```
Get-NetVitrualizationProviderRoute
```

```
PS C:\Users\Administrator\Desktop\NvgrePolicies> Get-NetVirtualizationProviderRoute


InterfaceIndex     : 34
DestinationPrefix  : 10.0.0.0/24
NextHop            : 0.0.0.0
Metric             : 255
```

Figure 23. Sample output for "Get-NetVirtualizationProviderRoute."

```
Get-NetVirtualizationLookupRecord
```

```
PS C:\Users\Administrator\Desktop\NvgrePolicies> Get-NetVirtualizationLookupRecord


CustomerAddress : 10.0.0.19
VirtualSubnetID : 6001
MACAddress      : 000002170012
ProviderAddress : 192.168.4.22
CustomerID      : {00000000-0000-0000-0000-000000000000}
Context         :
Rule            : TranslationMethodEncap
VMName          :
UseVmMACAddress : False
Type            : Static

CustomerAddress : 10.0.0.9
VirtualSubnetID : 5001
MACAddress      : 000002170002
ProviderAddress : 192.168.4.22
CustomerID      : {00000000-0000-0000-0000-000000000000}
Context         :
Rule            : TranslationMethodEncap
VMName          :
UseVmMACAddress : False
Type            : Static

CustomerAddress : 10.0.0.3
VirtualSubnetID : 5001
MACAddress      : 000001170002
ProviderAddress : 192.168.4.11
CustomerID      : {00000000-0000-0000-0000-000000000000}
Context         :
Rule            : TranslationMethodEncap
VMName          :
UseVmMACAddress : False
Type            : Static

CustomerAddress : 10.0.0.13
VirtualSubnetID : 6001
MACAddress      : 000001170012
ProviderAddress : 192.168.4.11
CustomerID      : {00000000-0000-0000-0000-000000000000}
Context         :
Rule            : TranslationMethodEncap
VMName          :
UseVmMACAddress : False
Type            : Static
```

Figure 24. Sample output for "Get-NetVirtualizationLookupRecord."

```
Get-NetVirtualizationCustomerRoute
```

```
PS C:\Users\Administrator\Desktop\NvgrePolicies> Get-NetVirtualizationCustomerRoute


RoutingDomainID   : {11111111-2222-3333-4444-000000006001}
VirtualSubnetID   : 6001
DestinationPrefix : 10.0.0.0/24
NextHop           : 0.0.0.0
Metric            : 255

RoutingDomainID   : {11111111-2222-3333-4444-000000005001}
VirtualSubnetID   : 5001
DestinationPrefix : 10.0.0.0/24
NextHop           : 0.0.0.0
Metric            : 255
```

Figure 25. Sample output for "Get-NetVirtualizationCustomerRoute."

c) Turn ON the VMs on “nvgre1” and “nvgre2.

d) Ping Blue2 from Blue1 and Red1 from Red2 to ensure that the VM network is pingable.

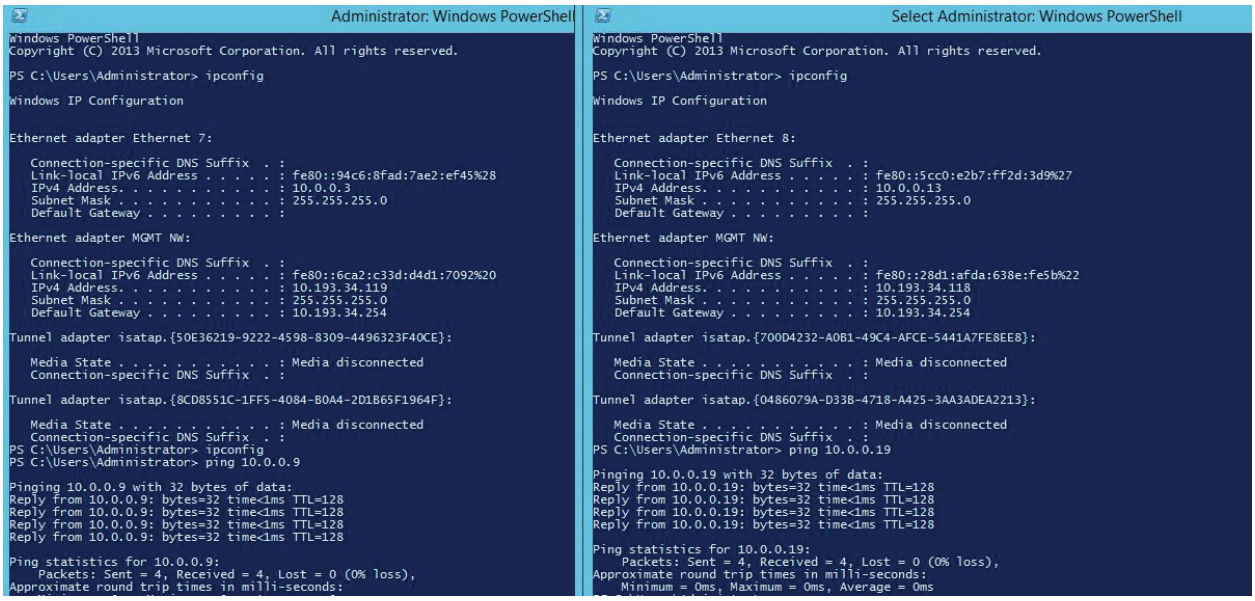
Administrator: Windows PowerShell
Windows PowerShell
Copyright (C) 2013 Microsoft Corporation. All rights reserved.
PS C:\Users\Administrator> ipconfig
Windows IP Configuration
Ethernet adapter Ethernet 7:
Connection-specific DNS Suffix . :
Link-local IPv6 Address . . . . . : fe80::94c6:8fad:7ae2:ef45%28
IPv4 Address. . . . . : 10.0.0.3
Subnet Mask . . . . . : 255.255.255.0
Default Gateway . . . . . :
Ethernet adapter MGMT NW:
Connection-specific DNS Suffix . :
Link-local IPv6 Address . . . . . : fe80::6ca2:c33d:d4d1:7092%20
IPv4 Address. . . . . : 10.193.34.119
Subnet Mask . . . . . : 255.255.255.0
Default Gateway . . . . . : 10.193.34.254
Tunnel adapter isatap.{50E36219-9222-4598-8309-4496323F40CE}:
Media State . . . . . : Media disconnected
Connection-specific DNS Suffix . :
Tunnel adapter isatap.{8CD8551C-1FF5-4084-B0A4-2D1B65F1964F}:
Media State . . . . . : Media disconnected
Connection-specific DNS Suffix . :
PS C:\Users\Administrator> ipconfig
PS C:\Users\Administrator> ping 10.0.0.9
Pinging 10.0.0.9 with 32 bytes of data:
Reply from 10.0.0.9: bytes=32 time<1ms TTL=128
Reply from 10.0.0.9: bytes=32 time<1ms TTL=128
Reply from 10.0.0.9: bytes=32 time<1ms TTL=128
Reply from 10.0.0.9: bytes=32 time<1ms TTL=128
Ping statistics for 10.0.0.9:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Select Administrator: Windows PowerShell
Windows PowerShell
Copyright (C) 2013 Microsoft Corporation. All rights reserved.
PS C:\Users\Administrator> ipconfig
Windows IP Configuration
Ethernet adapter Ethernet 8:
Connection-specific DNS Suffix . :
Link-local IPv6 Address . . . . . : fe80::5cc0:e2b7:ff2d:3d9%27
IPv4 Address. . . . . : 10.0.0.13
Subnet Mask . . . . . : 255.255.255.0
Default Gateway . . . . . :
Ethernet adapter MGMT NW:
Connection-specific DNS Suffix . :
Link-local IPv6 Address . . . . . : fe80::28d1:afda:638e:fe5b%22
IPv4 Address. . . . . : 10.193.34.118
Subnet Mask . . . . . : 255.255.255.0
Default Gateway . . . . . : 10.193.34.254
Tunnel adapter isatap.{70OD4232-A0B1-49C4-AFCE-5441A7FE8EE8}:
Media State . . . . . : Media disconnected
Connection-specific DNS Suffix . :
Tunnel adapter isatap.{0486079A-D33B-4718-A425-3AA3ADEA2213}:
Media State . . . . . : Media disconnected
Connection-specific DNS Suffix . :
PS C:\Users\Administrator> ping 10.0.0.19
Pinging 10.0.0.19 with 32 bytes of data:
Reply from 10.0.0.19: bytes=32 time<1ms TTL=128
Reply from 10.0.0.19: bytes=32 time<1ms TTL=128
Reply from 10.0.0.19: bytes=32 time<1ms TTL=128
Reply from 10.0.0.19: bytes=32 time<1ms TTL=128
Ping statistics for 10.0.0.19:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 0ms, Average = 0ms

Figure 26. Sample output for working ping.

Setup verification

After the basic network validation is performed ensuring that the NVGRE policies were set correctly, we can use any I/O tool to validate the multi-tenant topology.

a. Netperf, IOMeter or any IO tool can be used to verify the test setup.

b. Create a script using IxChariot I/O tool with the all the endpoint details. See the IxChariot Guide on how to create the test scripts.

c. For this POC, ultra high performance throughput script was used with 32 threads for each VM.

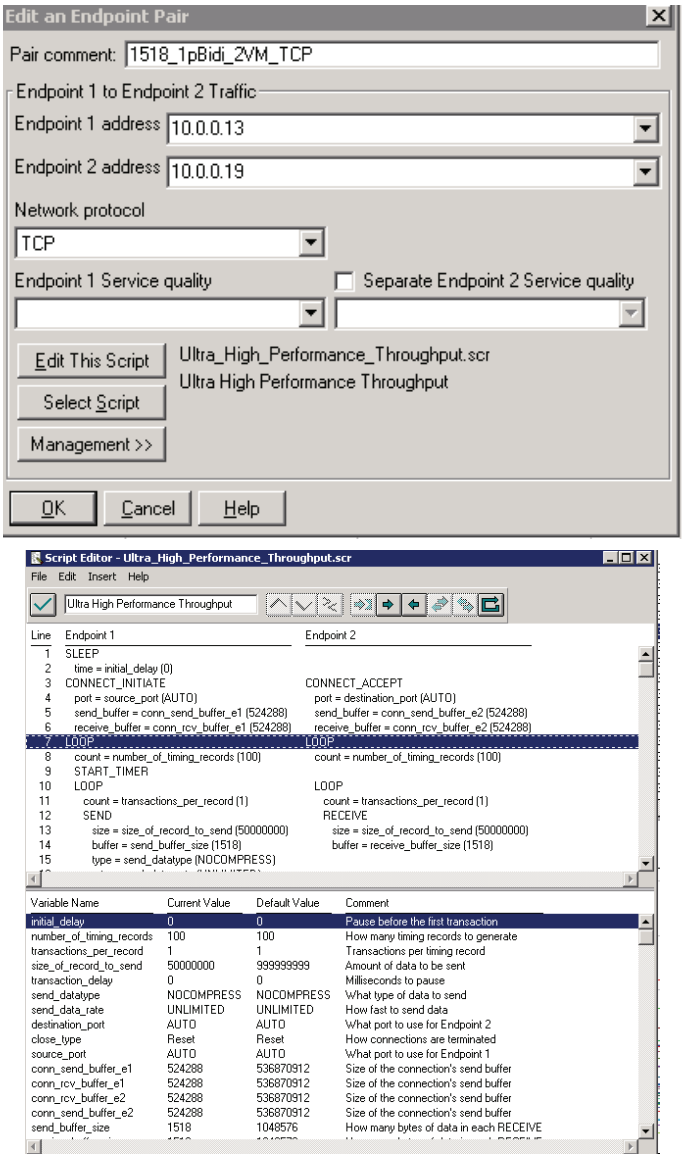
Edit an Endpoint Pair
Pair comment: 1518_1pBidi_2VM_TCP
Endpoint 1 to Endpoint 2 Traffic
Endpoint 1 address 10.0.0.13
Endpoint 2 address 10.0.0.19
Network protocol
TCP
Endpoint 1 Service quality
Separate Endpoint 2 Service quality
Edit This Script
Select Script
Management >>
Ultra_High_Performance_Throughput.scr
Ultra High Performance Throughput
OK
Cancel
Help
Script Editor - Ultra_High_Performance_Throughput.scr
File Edit Insert Help
Ultra High Performance Throughput
Line Endpoint 1 Endpoint 2
1 SLEEP
2 time = initial_delay (0)
3 CONNECT_INITIATE CONNECT_ACCEPT
4 port = source_port (AUTO) port = destination_port (AUTO)
5 send_buffer = conn_send_buffer_e1 (524288) send_buffer = conn_send_buffer_e2 (524288)
6 receive_buffer = conn_rcv_buffer_e1 (524288) receive_buffer = conn_rcv_buffer_e2 (524288)
7 LOOP LOOP
8 count = number_of_timing_records (100) count = number_of_timing_records (100)
9 START_TIMER
10 LOOP LOOP
11 count = transactions_per_record (1) count = transactions_per_record (1)
12 SEND RECEIVE
13 size = size_of_record_to_send (50000000) size = size_of_record_to_send (50000000)
14 buffer = send_buffer_size (1518) buffer = receive_buffer_size (1518)
15 type = send_datatype (NOCOMPRESS)
Variable Name Current Value Default Value Comment
initial_delay 0 0 Pause before the first transaction
number_of_timing_records 100 100 How many timing records to generate
transactions_per_record 1 1 Transactions per timing record
size_of_record_to_send 50000000 999999999 Amount of data to be sent
transaction_delay 0 0 Milliseconds to pause
send_datatype NOCOMPRESS NOCOMPRESS What type of data to send
send_data_rate UNLIMITED UNLIMITED How fast to send data
destination_port AUTO AUTO What port to use for Endpoint 2
close_type Reset Reset How connections are terminated
source_port AUTO AUTO What port to use for Endpoint 1
conn_send_buffer_e1 524288 536870912 Size of the connection's send buffer
conn_rcv_buffer_e1 524288 536870912 Size of the connection's send buffer
conn_rcv_buffer_e2 524288 536870912 Size of the connection's send buffer
conn_send_buffer_e2 524288 536870912 Size of the connection's send buffer
send_buffer_size 1518 1048576 How many bytes of data in each RECEIVE

Figure 27. Sample script for IxChariot.

d. Run the script to make sure the I/O run is successful.

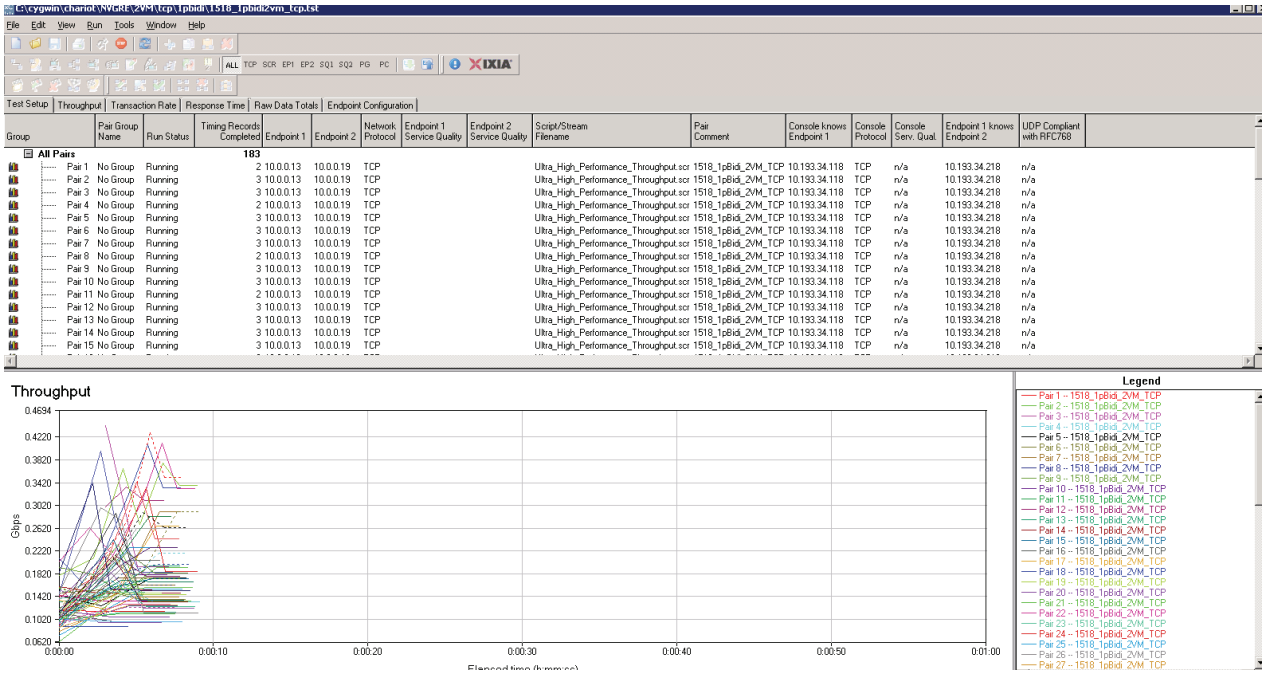
File Edit View Run Tools Window Help
Test Setup | Throughput | Transaction Rate | Response Time | Raw Data Totals | Endpoint Configuration
Group | Pair Group Name | Run Status | Timing Records Completed | Endpoint 1 | Endpoint 2 | Network Protocol | Endpoint 1 Service Quality | Endpoint 2 Service Quality | Script/Stream Filename | Pair Comment | Console knows Endpoint 1 | Console Protocol | Console Serv. Qual. | Endpoint 1 knows Endpoint 2 | UDP Compliant with RFC768
All Pairs 183
Pair 1 No Group Running 2 10.0.0.13 10.0.0.19 TCP Ultra_High_Performance_Throughput.scr 1518_1pBidi_2VM_TCP 10.193.34.118 TCP n/a 10.193.34.218 n/a
Pair 2 No Group Running 3 10.0.0.13 10.0.0.19 TCP Ultra_High_Performance_Throughput.scr 1518_1pBidi_2VM_TCP 10.193.34.118 TCP n/a 10.193.34.218 n/a
Pair 3 No Group Running 3 10.0.0.13 10.0.0.19 TCP Ultra_High_Performance_Throughput.scr 1518_1pBidi_2VM_TCP 10.193.34.118 TCP n/a 10.193.34.218 n/a
Pair 4 No Group Running 2 10.0.0.13 10.0.0.19 TCP Ultra_High_Performance_Throughput.scr 1518_1pBidi_2VM_TCP 10.193.34.118 TCP n/a 10.193.34.218 n/a
Pair 5 No Group Running 3 10.0.0.13 10.0.0.19 TCP Ultra_High_Performance_Throughput.scr 1518_1pBidi_2VM_TCP 10.193.34.118 TCP n/a 10.193.34.218 n/a
Pair 6 No Group Running 3 10.0.0.13 10.0.0.19 TCP Ultra_High_Performance_Throughput.scr 1518_1pBidi_2VM_TCP 10.193.34.118 TCP n/a 10.193.34.218 n/a
Pair 7 No Group Running 3 10.0.0.13 10.0.0.19 TCP Ultra_High_Performance_Throughput.scr 1518_1pBidi_2VM_TCP 10.193.34.118 TCP n/a 10.193.34.218 n/a
Pair 8 No Group Running 2 10.0.0.13 10.0.0.19 TCP Ultra_High_Performance_Throughput.scr 1518_1pBidi_2VM_TCP 10.193.34.118 TCP n/a 10.193.34.218 n/a
Pair 9 No Group Running 3 10.0.0.13 10.0.0.19 TCP Ultra_High_Performance_Throughput.scr 1518_1pBidi_2VM_TCP 10.193.34.118 TCP n/a 10.193.34.218 n/a
Pair 10 No Group Running 3 10.0.0.13 10.0.0.19 TCP Ultra_High_Performance_Throughput.scr 1518_1pBidi_2VM_TCP 10.193.34.118 TCP n/a 10.193.34.218 n/a
Pair 11 No Group Running 2 10.0.0.13 10.0.0.19 TCP Ultra_High_Performance_Throughput.scr 1518_1pBidi_2VM_TCP 10.193.34.118 TCP n/a 10.193.34.218 n/a
Pair 12 No Group Running 3 10.0.0.13 10.0.0.19 TCP Ultra_High_Performance_Throughput.scr 1518_1pBidi_2VM_TCP 10.193.34.118 TCP n/a 10.193.34.218 n/a
Pair 13 No Group Running 3 10.0.0.13 10.0.0.19 TCP Ultra_High_Performance_Throughput.scr 1518_1pBidi_2VM_TCP 10.193.34.118 TCP n/a 10.193.34.218 n/a
Pair 14 No Group Running 3 10.0.0.13 10.0.0.19 TCP Ultra_High_Performance_Throughput.scr 1518_1pBidi_2VM_TCP 10.193.34.118 TCP n/a 10.193.34.218 n/a
Pair 15 No Group Running 3 10.0.0.13 10.0.0.19 TCP Ultra_High_Performance_Throughput.scr 1518_1pBidi_2VM_TCP 10.193.34.118 TCP n/a 10.193.34.218 n/a
Throughput
Gbps
0.4694
0.4220
0.3820
0.3420
0.3020
0.2620
0.2220
0.1820
0.1420
0.1020
0.0620
0:00:00
0:00:10
0:00:20
0:00:30
0:00:40
0:00:50
0:01:00
Legend
Pair 1 -- 1518_1pBidi_2VM_TCP
Pair 2 -- 1518_1pBidi_2VM_TCP
Pair 3 -- 1518_1pBidi_2VM_TCP
Pair 4 -- 1518_1pBidi_2VM_TCP
Pair 5 -- 1518_1pBidi_2VM_TCP
Pair 6 -- 1518_1pBidi_2VM_TCP
Pair 7 -- 1518_1pBidi_2VM_TCP
Pair 8 -- 1518_1pBidi_2VM_TCP
Pair 9 -- 1518_1pBidi_2VM_TCP
Pair 10 -- 1518_1pBidi_2VM_TCP
Pair 11 -- 1518_1pBidi_2VM_TCP
Pair 12 -- 1518_1pBidi_2VM_TCP
Pair 13 -- 1518_1pBidi_2VM_TCP
Pair 14 -- 1518_1pBidi_2VM_TCP
Pair 15 -- 1518_1pBidi_2VM_TCP
Pair 16 -- 1518_1pBidi_2VM_TCP
Pair 17 -- 1518_1pBidi_2VM_TCP
Pair 18 -- 1518_1pBidi_2VM_TCP
Pair 19 -- 1518_1pBidi_2VM_TCP
Pair 20 -- 1518_1pBidi_2VM_TCP
Pair 21 -- 1518_1pBidi_2VM_TCP
Pair 22 -- 1518_1pBidi_2VM_TCP
Pair 23 -- 1518_1pBidi_2VM_TCP
Pair 24 -- 1518_1pBidi_2VM_TCP
Pair 25 -- 1518_1pBidi_2VM_TCP
Pair 26 -- 1518_1pBidi_2VM_TCP

Figure 28. Sample successful I/O output.

e. Please refer to Wireshark for detailed steps on how to capture packet. This is an optional step for verification.

f. Wireshark was used to capture some of the packets to take a look at the contents. On the capture for NVGRE enabled, "Generic Routing Encapsulation (Transparent Ethernet Bridging)" header was seen in the Ethernet packet.

g. For traffic with no NVGRE policy, no Generic Routing Encapsulation (Transparent Ethernet Bridging)" header was seen in the Ethernet packet.

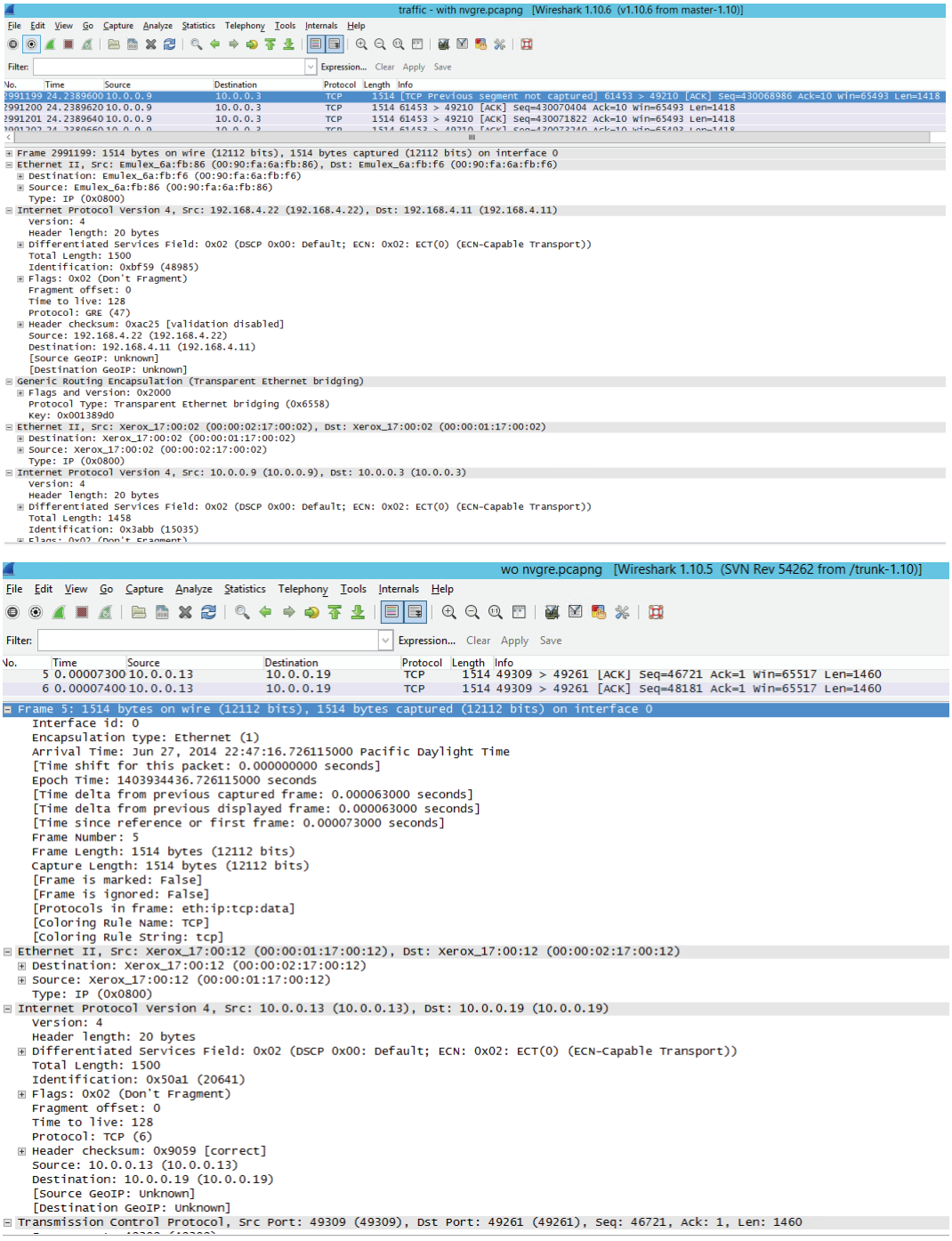
traffic - with nvgre.pcapng [Wireshark 1.10.6 (v1.10.6 from master-1.10)]
wo nvgre.pcapng [Wireshark 1.10.5 (SVN Rev 54262 from /trunk-1.10)]

Figure 29. Sample wireshark captures.

Remove NVGRE policy

If there are any changes required to the existing configuration, like adding/removing any VM or changing any MAC address or changing VSID, we need to remove the existing policy before enabling the new changes. Please make sure the VMs are shut down before executing the remove policy script.

1. To remove the NVGRE policy, create a RemovePolicy.ps1 script on "nvgre1" and "nvgre2."
2. Add `"Remove-NetVirtualizationCustomerRoute," "Remove-NetVirtualizationLookupRecord," "Remove-NetVirtualizationProviderAddress," "Remove-NetVirtualizationProviderRoute"` to RemovePolicy.ps1 script for both the hosts.
3. Assign "0" to VSID to each VM on both the hosts.
4. Add the lines below to RemovePolicy.ps1 on nvgre1.

```
Get-VMNetworkAdapter "Blue1" | where {$_.MacAddress -eq "000001170002"} | Set-VMNetworkAdapter
-VirtualSubnetID 0

Get-VMNetworkAdapter "Red1" | where {$_.MacAddress -eq "000001170012"} | Set-VMNetworkAdapter
-VirtualSubnetID 0
```

5. Add the lines below to RemovePolicy.ps1 on nvgre2.

```
Get-VMNetworkAdapter "Blue2" | where {$_.MacAddress -eq "000002170002"} | Set-VMNetworkAdapter
-VirtualSubnetID 0

Get-VMNetworkAdapter "Red2" | where {$_.MacAddress -eq "000002170012"} | Set-VMNetworkAdapter
-VirtualSubnetID 0
```

6. Shutdown all the VMs and then run the script on each host.

Phase 2 – Two hosts connected back to back

The steps for this phase are exactly the same as Phase1. If a 10Gbps switch is not available for implementing the NVGRE feature, this phase can be implemented instead of phase 1.

Conclusion

In summary, the overview of the hardware and software components needed to successfully deploy a NVGRE configuration was introduced as a part of this application note. As it can be seen, configuring and deploying NVGRE on a Windows Server 2012 R2 with the Emulex OCe14102 adapter is fairly simple. The steps and powershell commands go hand-in-hand for deployment. When evaluating the deployment, it is important to understand the method used for configuration and the environment where it is deployed. Please refer to Microsoft for detailed deployment steps of Hybrid Cloud with NVGRE using VMM.

Appendix A: NVGRE policy powershell script used for the above configuration

```
##########################################################################################################
#################
 # Blue Virtual Network Information
 #
 # RoutingDomainID="{11111111-2222-3333-4444-000000005001}"
 # VirtualSubnetID]=5001
 # (Both RDID and VSID are defined by administrators, MUST be unique in the datacenter)
 #
 # [Customer Addresses]
 # VM Name Host VSID CA PA MAC DefaultGW
 # ------------------------------------------------------------------------------
 # Blue1 Host1 5001 10.0.0.3 192.168.4.11 00-00-01-17-00-02 10.0.0.1
 # Blue2 Host2 5001 10.0.0.9 192.168.4.22 00-00-02-17-00-02 10.0.0.1
 #
 # [Customer Routes]
 # DestPrefix NextHopGW Note
 # 10.0.1.0/24 0.0.0.0 Onlink route for Blue subnet
# Add the locator records for Blue subnet
New-NetVirtualizationLookupRecord -VirtualSubnetID “5001” -CustomerAddress “10.0.0.3”
-ProviderAddress “192.168.4.11” -MACAddress “000001170002” -Rule “TranslationMethodEncap”
New-NetVirtualizationLookupRecord -VirtualSubnetID “5001” -CustomerAddress “10.0.0.9”
-ProviderAddress “192.168.4.22” -MACAddress “000002170002” -Rule “TranslationMethodEncap”
 # Add the customer route records for Blue subnet
 New-NetVirtualizationCustomerRoute -RoutingDomainID “{11111111-2222-3333-4444-000000005001}”
-VirtualSubnetID “5001” -DestinationPrefix “10.0.0.0/24” -NextHop “0.0.0.0” -Metric 255
##########################################################################################################
#################
 #Red Virtual Network Information
 #
 # RoutingDomainID="{11111111-2222-3333-4444-000000006001}"
 # VirtualSubnetID=6001
 # (Both RDID and VSID are defined by administrators, MUST be unique in the datacenter)
 #
 # [Customer Addresses]
```

```
 # VM Name Host VSID CA PA MAC DefaultGW

 # ---------------------------------------------------------------------------------------

 # Red1 Host1 6001 10.0.0.13 192.168.4.11 00-00-01-17-00-12 10.0.0.1

 # Red2 Host2 6001 10.0.0.19 192.168.4.22 00-00-02-17-00-12 10.0.0.1

 # [Customer Routes]

 # DestPrefix NextHopGW Note

 # 10.0.0.0/24 0.0.0.0 Onlink route for Red subnet

 # Add the locator records for Red subnet

New-NetVirtualizationLookupRecord -VirtualSubnetID “6001” -CustomerAddress “10.0.0.13”
-ProviderAddress “192.168.4.11” -MACAddress “000001170012” -Rule “TranslationMethodEncap”

New-NetVirtualizationLookupRecord -VirtualSubnetID “6001” -CustomerAddress “10.0.0.19”
-ProviderAddress “192.168.4.22” -MACAddress “000002170012” -Rule “TranslationMethodEncap”

# Add the customer route records for Red subnet

New-NetVirtualizationCustomerRoute -RoutingDomainID “{11111111-2222-3333-4444-000000006001}”
-VirtualSubnetID “6001” -DestinationPrefix “10.0.0.0/24” -NextHop “0.0.0.0” -Metric 255

#########################################################################################
#################

# [2] Configure the Host Provider Addresses and Routes required for this setup

# [Host PA Address & Route information required by the VM policy]

# Host Hostname {PA’s} {VM:VirtualSubnetID} ==> Set on the VMNetworkAdapter on each host

# ---------------------------------------------------------------------------------------
----------------------------

# Host1 example-host1 192.168.4.11 {Blue1:5001, Red1:6001}

# Host2 example-host2 192.168.4.22 {Blue2:5001, Red2:6001}

 # [2-2] Host2

 # (a) Configure Provider Address and Route:

#Get-NetAdapter

New-NetVirtualizationProviderAddress -InterfaceIndex 34 -ProviderAddress 192.168.4.11
-PrefixLength 24

 # (b) Set VirtualSubnetID on the VM network port

Get-VMNetworkAdapter “Blue1” | where {$_.MacAddress -eq “000001170002”} | Set-VMNetworkAdapter
-VirtualSubnetID 5001

Get-VMNetworkAdapter “Red1” | where {$_.MacAddress -eq “000001170012”} | Set-VMNetworkAdapter
-VirtualSubnetID 6001

# ProviderRoute

New-NetVirtualizationProviderRoute -DestinationPrefix “10.0.0.0/24” -InterfaceIndex 34 -NextHop
“0.0.0.0” -Metric 255
```

Appendix B: NVGRE policy removal powershell script used for the above configuration

```
Get-NetVirtualizationCustomerRoute

Get-NetVirtualizationGlobal

Get-NetVirtualizationLookupRecord

Get-NetVirtualizationProviderAddress

Get-NetVirtualizationProviderRoute

Remove-NetVirtualizationCustomerRoute

Remove-NetVirtualizationLookupRecord

Remove-NetVirtualizationProviderAddress

Remove-NetVirtualizationProviderRoute

Get-VMNetworkAdapter “Blue1” | where {$_.MacAddress -eq “000001170002”} | Set-VMNetworkAdapter
-VirtualSubnetID 0

Get-VMNetworkAdapter “Red1” | where {$_.MacAddress -eq “000001170012”} | Set-VMNetworkAdapter
-VirtualSubnetID 0

Get-NetVirtualizationCustomerRoute

Get-NetVirtualizationGlobal

Get-NetVirtualizationLookupRecord

Get-NetVirtualizationProviderAddress

Get-NetVirtualizationProviderRoute
```

More information

The Implementer's Lab website
www.implementerslab.com

Hyper-V Network Virtualization technical details
http://technet.microsoft.com/enus/library/jj134174.aspx

Simple Hyper-V Network Virtualization Demo script
http://gallery.technet.microsoft.com/scriptcenter/Simple-Hyper-V-Network-d3efb3b8

Hyper-V Network Virtualization Gateway Architectural Guide
http://technet.microsoft.com/en-us/library/jj618319.aspx

Simple Hyper-V Network Virtualization Script with Gateway
http://gallery.technet.microsoft.com/scriptcenter/Simple-Hyper-V-Network-6928e91b

NVGRE IETF draft
http://tools.ietf.org/html/draft-sridharan-virtualization-nvgre-02

To help us improve our documents, please provide feedback at
implementerslab@emulex.com

Some of these products may not be available in the U.S. Please contact your supplier for more information.

EMULEX®

World Headquarters 3333 Susan Street, Costa Mesa, CA 92626 +1 714 662 5600
Bangalore, India +91 80 40156789 | **Beijing, China** +86 10 84400221
Dublin, Ireland +35 3 (0) 1 652 1700 | **Munich, Germany** +49 (0) 89 97007 177
Paris, France +33 (0) 158 580 022 | **Tokyo, Japan** +81 3 5325 3261 | **Singapore** +65 6866 3768
Wokingham, United Kingdom +44 (0) 118 977 2929 | **Brazil** +55 11 3443 7735

www.emulex.com

©2014 Emulex, Inc. All rights reserved.
This document refers to various companies and products by their trade names. In most cases, their respective companies claim these designations as trademarks or registered trademarks. This information is provided for reference only. Although this information is believed to be accurate and reliable at the time of publication, Emulex assumes no responsibility for errors or omissions. Emulex reserves the right to make changes or corrections without notice. This document is the property of Emulex and may not be duplicated without permission from the Company.